Exploration of Continual Learning Paradigm for Large-Scale Speaker Verification Systems

A Project Report

submitted by

AKASH VERMA

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

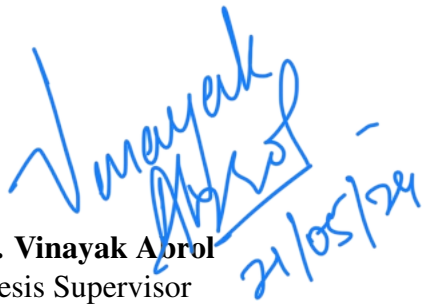
COMPUTER SCIENCE ENGINEERING
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

May 2024

THESIS CERTIFICATE

This is to certify that the thesis titled **Exploration of Continual Learning Paradigm for Large-Scale Speaker Verification Systems**, submitted by **Akash Verma**, to the INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY, DELHI, for the award of the degree of **Master of Technology**, in Computer Science Engineering with specialization in AI, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.



Dr. Vinayak Abrol
Thesis Supervisor
Assistant Professor
Dept. of Computer Science Engineering
IIIT Delhi, 110020

Place: New Delhi
Date: May 21, 2024

ACKNOWLEDGEMENTS

I am profoundly grateful to many individuals and institutions whose support and encouragement have been indispensable throughout my journey in completing this thesis.

First and foremost, I would like to extend my heartfelt gratitude to the **Center for Artificial Intelligence (CAI) at IIIT Delhi** for providing the essential computing resources that were crucial for my research. Their support significantly facilitated the complex computations required for this work.My sincere thanks also go to the **Cross Caps Lab** for providing the infrastructure and space that enabled a conducive environment for my research activities. I am especially indebted to **Dr. Vinayak Abrol**, whose guidance and supervision have been invaluable. His dedication, knowledge, and inspirational leadership have been a constant source of motivation and have greatly contributed to the depth and quality of this thesis.

I would like to acknowledge my family and friends and **Vishal Kumar**, the PHD colleague for their unwavering support and encouragement. Their belief in my abilities and their emotional support have been vital in overcoming the challenges encountered along this journey.

My MTech thesis journey began in January 2023 and was marked by numerous personal and academic challenges. The process involved extensive literature review to understand the current state of the art in Continual Learning and Speaker Verification Systems, and practical implementation posed its own set of challenges, especially with handling a complex codebase.Despite these hurdles, the journey has been incredibly rewarding. One of the major milestones was securing a placement at **Mercedes-Benz Research & Development India**, which stands as a testament to the skills and knowledge acquired during this period.

Thank you all for being a part of this incredible journey.

ABSTRACT

KEYWORDS: Speaker Verification; Continual Learning; Catastrophic Forgetting; Stability v/s Plasticity Dilemma; Audio Processing; Class Incremental Training; Speech and Audio Applications

Speaker verification (SV) focuses on confirming or denying the claimed identity of a speaker. It is a one-to-one comparison between the test utterance and the claimant's stored reference voice sample. This process is commonly used in security applications for access control and authentication. One of the popular approaches to building SV systems is to use a speaker identification embedding/encoder model as a feature extractor that is pre-trained on a large number of known speakers. While Deep Neural Networks (DNNs) as encoders have achieved significant benchmarks in SV, they often fail to adapt to new data due to catastrophic forgetting. In such cases, Continual Learning, a technique involving incremental learning or task-based learning, empowers the model to retain previously learned information and handle diverse new information when available. However, adapting SV models to handle new classes without complete retraining remains challenging. Using the Voxceleb2 dataset, one of the largest datasets with over 6000 speakers from 145 nationalities, this study explores the implementation of different settings to develop models flexible enough to generalize to distributional shifts in the data without requiring complete retraining. The research investigates how continual learning techniques can effectively help speech verification systems adapt and scale to maintain model performance while accommodating a vast and dynamic array of classes. Through empirical evaluations on benchmark datasets and simulated real-world scenarios, the study assesses the efficacy of various continual learning approaches in mitigating forgetting and adapting to new classes in speaker verification tasks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
CL	Continual Learning
RL	Reinforcement Learning
CNN	Convolutional Neural Network
DNN	Deep Neural Network
EER	Equal Error Rate
EWC	Elastic Weight Consolidation
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
LSTM	Long Short-Term Memory
min DCF	Minimum Detection Cost Function
ML	Machine Learning
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SV	Speaker Verification
TDNN	Time Delay Neural Network
VoxCeleb	A large-scale speaker verification dataset

NOTATION

θ	Model parameters
$\mathcal{L}$	Loss function
$\mathcal{D}$	Dataset
$\mathcal{D}_{\textbf{base}}$	Base dataset
$\mathcal{D}_{\textbf{new}}$	New dataset
Ω	Importance of parameters
$E_{\textbf{new}}$	Error rate for new data
$E_{\textbf{base}}$	Error rate for base data
λ	Regularization parameter
$\mathcal{T}$	Task or subset of data used in incremental learning
$\hat{y}$	Predicted output
y	Actual output
$\mathcal{M}$	Memory buffer for storing past data
$f(\cdot)$	Feature extraction function
$\mathcal{N}$	Number of classes or speakers
Δ	Change in a variable
$\mathbf{x}$	Input feature vector
$\mathbf{h}$	Hidden state in neural networks
$\mathbf{W}$	Weight matrix in neural networks
b	Bias term in neural networks
$\sigma(\cdot)$	Activation function

Most of the notations used are specified here. In terms of clash each and every notation has been defined where they have been used.

CHAPTER 1

INTRODUCTION

1.1 Background

Human voices carry unique characteristics due to individual anatomical structures like the size of the larynx, the shape of the vocal tract, and distinct speaking styles such as accents. These unique vocal traits form the basis for voice biometrics, which are used in identification and authentication technologies. Speaker verification (SV) involves verifying the claimed identity of a speaker using their voice. This is a crucial task in speech processing with extensive real-world applications, such as voice-based device authentication for smartphones, smart home systems, and vehicles, as well as in combating telecommunication fraud by identifying scammers.The process of speaker verification can be broken down into three primary stages: training, enrollment, and evaluation. During the training phase, a large dataset of speech samples from numerous speakers is utilized to develop a speaker embedding extractor and a scoring function. This stage is crucial as it lays the foundation for the system's ability to distinguish between different speakers. In the enrollment stage, a few utterances from a specific speaker are used to create a unique speaker model. This model serves as the reference against which future utterances will be compared. Finally, in the evaluation stage, a new utterance is compared to the enrolled speaker model to verify the claimed identity. If the evaluation score exceeds a predefined threshold, the identity claim is accepted; otherwise, it is rejected.

1.1.1 Types of Speaker Recognition Methods

Speaker recognition encompasses various techniques that are differentiated by their recognition policies and environmental conditions. This section delves into the different sub-domains of speaker recognition, categorized according to the specific recognition criteria they employ. Speaker verification can be divided into text-dependent and text-independent categories. In text-dependent SV, the speech content during the enrollment

and evaluation stages is restricted to specific phrases. Conversely, in text-independent SV, the speech content is not restricted and can be entirely spontaneous. Text-dependent systems are becoming obsolete as users often consider text-dependency as a limitation. A brief overview about the types of Speaker Recognition strategies can be seen in Figure 1.1.

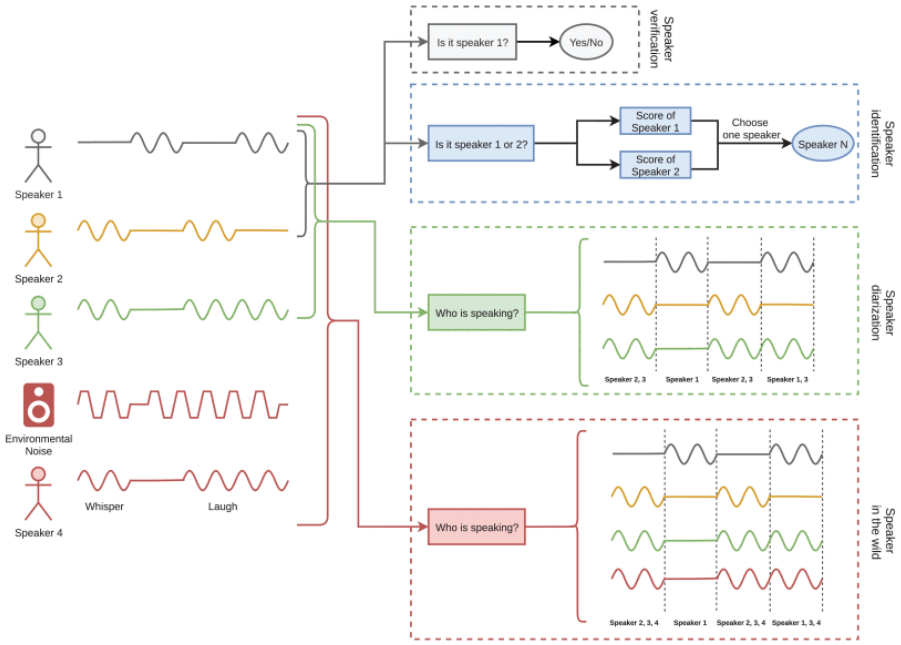


Figure 1.1: Different Types of Speaker Recognition Strategies.
Ohi *et al.* (2021)

1. **Speaker Verification**

   Speaker verification focuses on confirming or denying a user's claimed identity based on their voice. Conceptually, this task can be framed as a binary classification problem. A deep learning-based speaker verification system aims to determine if a given voice sample matches the claimed identity. The objective is to train a model with parameters w_f that minimizes the loss function L. This can be formally expressed as:

   $$L(f_{w_f}(x), y),$$

   here, $f_{w_f}(\cdot)$ represents a deep neural network model parameterized by w_f, which produces predictions $\hat{y}$. The goal is to adjust w_f such that the loss $L(\cdot, \cdot)$ is minimized, thereby optimizing the model's performance in distinguishing between authentic and impostor voice samples.

2. **Speaker Identification**

   Speaker identification involves identifying the speaker of a given utterance from a predefined set of registered speakers. Unlike speaker verification, which is binary, speaker identification is a multi-class classification problem. The system must determine which speaker from a set of n speakers produced a given speech sample. Mathematically, this can be described by the following expression:

   $$L\left(\arg\max_{y}\{f_1 w_{f1}(x), f_2 w_{f2}(x), \ldots, f_n w_{fn}(x)\}, y\right)$$

In this equation, $f_i w_{fi}(\cdot)$ denotes the deep neural network classifier that estimates the probability of the i-th speaker being the source of the utterance, where $i \leq n$. Speaker identification systems can also be seen as an extension of speaker verification systems, as they can be constructed by combining multiple verification models to handle the multi-class nature of the identification problem.

3. **Speaker Diarization**

   Speaker diarization is another variant of speaker identification, primarily used in scenarios where it is necessary to determine "who spoke when." This is particularly useful in automated speech transcription systems, which need to segment and label the speech from multiple speakers within a single audio stream. Unlike standard speaker identification, which typically deals with identifying a single speaker per utterance, speaker diarization must handle overlapping speech from multiple speakers, making it a multi-output/multi-label classification problem. Importantly, speaker diarization systems usually operate without a prior enrollment process and work unsupervised to distinguish and label speakers.

4. **Speaker Verification in the Wild**

   Speaker verification "in the wild" refers to the application of speaker recognition techniques in real-world scenarios where the conditions are uncontrolled and unpredictable. This includes environments with various types of background noise, reverberation, channel distortions, and other acoustic interferences. In-the-wild speaker recognition tasks combine elements of verification, identification, and diarization, requiring robust models capable of handling diverse and corrupted audio inputs. The datasets used for these tasks often contain recordings with noise, laughter, music, echo, and cross-talks. Addressing the challenges posed by these conditions is a current focus of research, aiming to develop systems that can perform reliably in such unpredictable environments.

   By exploring these distinct sub-domains, we gain a comprehensive understanding of the different approaches and challenges in speaker recognition, laying the groundwork for further advancements and applications in this field. Ohi *et al.* (2021)

1.1.2 Neural Based Models for Speaker Verification

Deep learning has revolutionized speaker recognition by enabling systems to process speech data through two primary input patterns: directly processing raw sound waves or utilizing pre-processed data. This section explores these methodologies, their underlying principles, and their applications in modern speaker recognition architectures. The approaches are listed below:

1. **Processing Raw Sound Waves** In the first approach, speaker recognition models are designed to process raw sound waves directly. This method involves feeding unprocessed audio data into the system, which may require normalization based on the sound wave's characteristics or predefined thresholds. Recent architectures

like SincNet, RawNet, and AM-MobileNet exemplify this approach. These models are trained using raw speech inputs, capturing intricate details of the audio signals that might be lost during pre-processing. By handling raw sound waves, these systems can potentially leverage the full spectrum of audio information for more accurate speaker recognition.

2. **Utilizing Pre-Processed Data** Despite the advancements in raw audio processing, the majority of speaker recognition systems still rely on pre-processing techniques. These techniques convert the time-domain speech signals into representations that juxtapose time and frequency information. This transformation helps in capturing the essential features of speech more effectively, facilitating the recognition process. Pre-processing methods typically involve converting continuous time-domain signals into shorter, manageable segments known as frames.

3. **Segmentation of Speech Data** Speech data, inherently a continuous time signal, is generally segmented into shorter frames during training. In speaker recognition terminology, an "utterance" refers to a segment of sound that typically contains a single word or a brief phrase. In contrast, a "frame" is a much shorter segment of speech, ranging from 20 to 500 milliseconds, which may contain a complete utterance or just a portion of it. Recognizers often use these frames as input, as they encapsulate crucial phoneme information necessary for accurate recognition.

These approaches often rely on Feature Extraction Techniques. There are various Feature Extraction techniques available for processing the Raw Audio or segments of the speech data. An overview of these techniques are listed below:

Feature Extraction Techniques

Pre-processing speech data before feeding it into deep learning models is a common practice to enhance the efficiency and accuracy of recognition. The three primary types of pre-processing observed in speaker recognition systems are:

1. **Spectrogram:**This method involves converting the time-domain signal into a visual representation that shows how the signal's frequency content varies over time. Spectrograms are widely used because they provide a comprehensive view of the speech signal's frequency components.

2. **Mel-Filterbank:** The mel-filterbank approach applies a series of filters to the speech signal, spaced according to the mel scale, which mimics human auditory perception. This method captures the essential frequency characteristics of speech, facilitating more effective feature extraction.

3. **Mel-Frequency Cepstral Coefficients (MFCC):** MFCCs are a popular choice for speech and speaker recognition tasks. This technique involves taking the Fourier transform of the signal, applying a mel-scale filterbank, and then computing the logarithm of the energy at each filter output. The final step involves performing a discrete cosine transform to decorrelate the coefficients, resulting in a compact representation of the speech signal's spectral properties. Kaddoura and Vadlamudi (2016)

During both the enrollment and testing phases, equal-length speech frames are fed into the recognizer. This consistency ensures that the system can effectively compare and analyze the speech patterns. Pre-processing plays a crucial role in this process, transforming raw audio data into formats that deep learning models can efficiently process.Speaker verification systems generally follow two main architectures as shown in Figure 1.2: stage-wise and end-to-end.

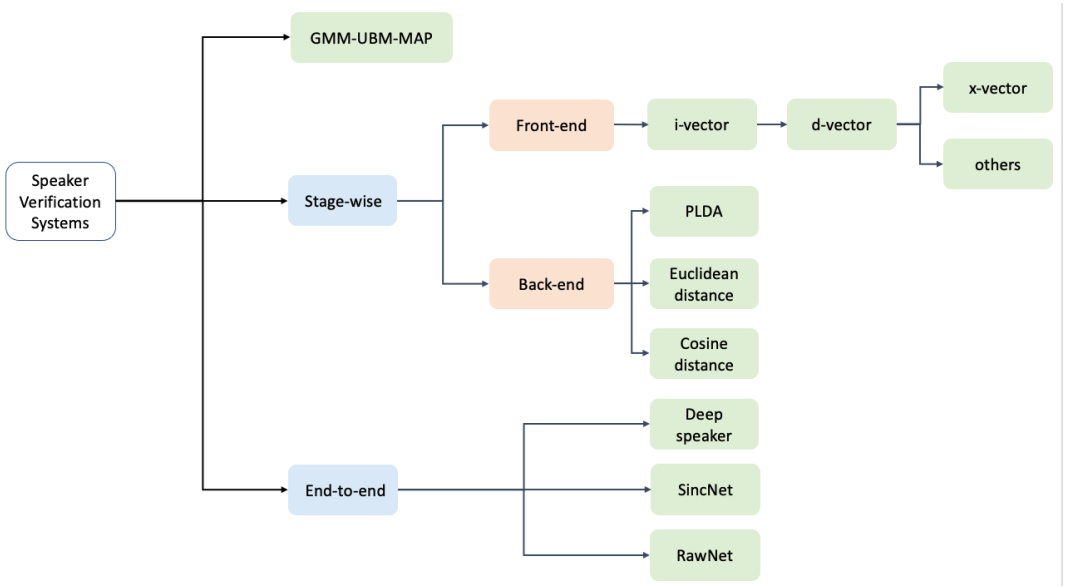


Figure 1.2: Speaker Verification Architectures.
Zhu (2022)

Stage-wise Systems: A typical stage-wise system comprises a front-end module for speaker representation learning and a back-end module for comparing speaker representations. These modules are trained separately. The workflow of such systems includes:

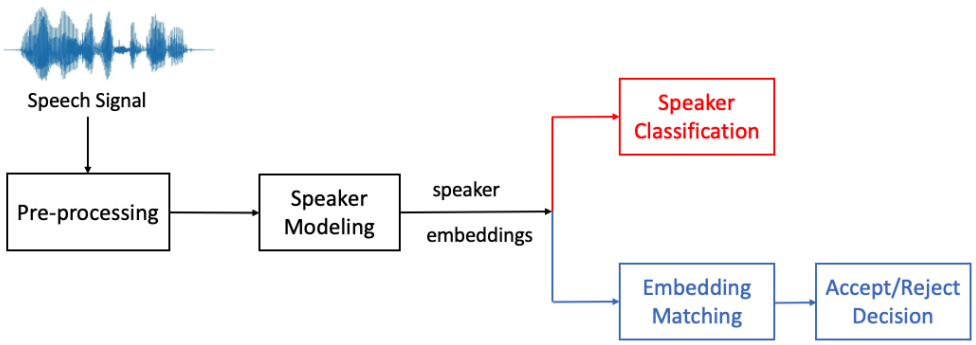


Figure 1.3: Workflow of Speaker Verification Systems.
Zhu (2022)

1. **Pre-processing:** This involves voice activity detection to eliminate non-speech segments and speech parameterization, converting speech signals into feature vectors using signal processing techniques. Common features include filterbanks and Mel-frequency Cepstral Coefficients (MFCC).

2. **Speaker Modeling:** Feature vectors are used to generate speaker representations or "embeddings." Traditional statistical methods like Gaussian Mixture Models (GMM) have been widely used, but recent advancements in deep learning have introduced models such as Time-Delay Neural Networks (TDNN), Long Short-Term Memory networks (LSTM), and Residual Networks (ResNet).

3. **System Workflow:** The workflow is illustrated in Figure 1.3. In the training stage, speaker embeddings are learned via a classification task. Post-training, classification layers are removed, and the embeddings are used to train an embedding matching model. During enrollment and evaluation, the speaker embeddings undergo the embedding matching process to compute similarity scores, which are then used to make verification decisions.

End-to-end Systems: End-to-end systems integrate the speaker modeling and embedding matching modules, optimizing them simultaneously during the training stage. These systems take speech utterances as input and directly produce similarity scores for the utterance pairs, as shown in Figure 1.3. Ohi *et al.* (2021)

1.1.3 Current Challenges in SV Systems

Despite the numerous advantages of Speaker Verification (SV) systems, they face several significant challenges that hinder their effectiveness and scalability.

1. **Vulnerability to Spoofing Attacks:** One of the primary challenges is the susceptibility to spoofing attacks, where an imposter attempts to mimic the voice of a legitimate user to gain unauthorized access. Attackers can use various techniques such as voice synthesis or replay attacks to deceive the system.

2. **Variability Due to Environmental Conditions:** SV systems often encounter variability in voice recordings due to differing environmental conditions. Factors such as background noise, recording equipment, and the physical condition of the speaker (e.g., illness, stress) can affect the accuracy of voice verification.

3. **Requirement for Large, Labeled Datasets:** Developing robust SV models requires large datasets with accurately labeled voice samples. Collecting and annotating such datasets is resource-intensive and time-consuming. Additionally, the diversity in accents, languages, and speaking styles further complicates the dataset requirements.

4. **Computationally Expensive Retraining:** Traditional SV models require retraining from scratch whenever new speakers are added to the system. This process

> is computationally expensive and impractical for large-scale applications where the speaker pool is dynamic and continuously expanding. The need for frequent retraining can lead to significant downtime and resource consumption, limiting the system's scalability. Mittal and Dua (2022)

In summary, speaker verification systems are crucial for secure and reliable voice-based authentication. The development of deep learning techniques has significantly enhanced the effectiveness of these systems, enabling more accurate and robust speaker verification in various applications.This chapter sets the foundation for understanding the mechanisms and architectures of speaker verification systems, which will be further explored in the subsequent chapters and sections of this thesis.

1.2 Problem Statement and Objectives

Our problem statement is majorly defined in aura of below 3 main Issues which Speaker Verification system faces. A detailed explanation of the issues are as follows

1.2.1 Issues with Catastrophic Forgetting in SV Systems

The current SV Systems are build for stationery data. They do not deal with real time new data. Catastrophic forgetting is a major issue in SV systems. It occurs when a model, after being updated with new data, loses information about previously learned speakers. This is particularly problematic in dynamic environments where new speakers are frequently introduced. The model's inability to retain previously acquired knowledge while learning new information leads to a decline in overall performance. Addressing catastrophic forgetting is crucial for the development of SV systems that can operate effectively in real-world, ever-changing scenarios.

1.2.2 Need for Continual Learning to Handle Dynamic and Expanding Datasets

To manage dynamic and expanding datasets, there is a pressing need for continual learning techniques in SV systems. Continual learning allows the model to adapt to new

speakers incrementally, without the necessity of complete retraining. This capability is essential for maintaining the system's efficiency and scalability. By incorporating continual learning, SV systems can seamlessly integrate new data, preserve past knowledge, and ensure robust performance across diverse and evolving datasets.

1.2.3 Focus on Class Incremental Strategy and Distributional Shifts

This thesis will focus on employing a class incremental strategy to address the challenges posed by continually expanding datasets. Class incremental learning involves introducing new classes (speakers) incrementally while retaining the knowledge of previously learned classes. This approach is particularly relevant for SV systems, where new speakers are continually added.In addition to class incremental learning, the research will address the issue of distributional shifts, which occur when the statistical properties of the training data change over time. Distributional shifts can degrade the performance of SV systems if not properly managed. The study will explore techniques to handle such shifts, ensuring that the SV models remain robust and effective despite changes in the data distribution.

By focusing on these areas, the thesis aims to develop advanced SV systems capable of continual learning, efficient integration of new speakers, and robust performance in the face of dynamic and shifting datasets.Hence, Thesis statement -
"This thesis delves into the realm of continual learning for large-scale speaker verification systems, aiming to optimize strategies that effectively address the challenges posed by evolving speaker datasets, mitigate the risk of catastrophic forgetting, and foster adaptability for sustained performance excellence in dynamic environments."

1.3 Thesis structure and Outline

This thesis is organized into several chapters, each addressing different aspects of the research on continual learning for large-scale speaker verification systems. The structure is designed to provide a comprehensive understanding of the challenges, methodologies, experimental results, and implications of this study. Below is an outline of the thesis:

Chapter 1: Introduction

1. **Background:** An overview of speaker verification (SV) systems, their importance, and applications. This section also introduces the concepts of continual learning and the problem of catastrophic forgetting in SV systems.

2. **Problem Statement and Objectives:** A detailed discussion of the key issues faced by current SV systems, such as catastrophic forgetting and the need for continual learning to handle dynamic and expanding datasets. The thesis statement is presented: ***"This thesis delves into the realm of continual learning for large-scale speaker verification systems, aiming to optimize strategies that effectively address the challenges posed by evolving speaker datasets, mitigate the risk of catastrophic forgetting, and foster adaptability for sustained performance excellence in dynamic environments."***

3. **Thesis Structure and Outline:** An overview of the organization of the thesis, detailing the contents of each chapter.

Chapter 2: Literature Review

1. **Speaker Verification Systems:** A review of traditional and existing methods for SV, including statistical models and deep neural network methods.

2. **Continual Learning:** An exploration of the continual learning paradigm, including related learning paradigms, formal definitions, scenarios, and various strategies.

3. **Application in Speaker Verification:** Discussion on how continual learning can be applied to SV, its advantages, challenges, and case studies.

Chapter 3: Proposed Work

1. **Problem Statement:** Reiteration of the main problem addressed in the thesis in a depthwise detailed manner addressing the key problem.

2. **Datasets:** Description of the datasets used (e.g., VoxCeleb2) and the baseline performance of existing models.

3. **Methodology:** Detailed explanation of the proposed class-incremental strategy, data splits, and continual learning strategies such as Elastic Weight Consolidation (EWC) and Replay methods.

4. **Experimental Setup:** Overview of the ECAPA-TDNN model and its enhancements. Presentation of the baseline results obtained from the ECAPA-TDNN model on various test sets. Explanation of the metrics used for evaluation, including Equal Error Rate (EER) and minimum Detection Cost Function (min DCF).

5. **Results and Discussion:** Detailed analysis of the experimental results, comparison of different continual learning strategies, and discussion of their implications.

Chapter 4: Conclusion

1. **Conclusion:** Discusses the advantages and disadvantages of the implemented methods and their impact on speaker verification systems.
2. **Summary:** Summarizes the key findings of the research, highlighting the effectiveness of the proposed continual learning strategies.
3. **Future Scope:** Outlines potential areas for future research, including advanced continual learning techniques, real-time continual learning, handling imbalanced and noisy data, cross-domain adaptation, integration with other modalities, and scalable deployment.

This structured approach ensures a thorough exploration of the continual learning paradigm for large-scale speaker verification systems, providing valuable insights and setting the stage for future advancements in the field.

CHAPTER 2

Literature Review

2.1 Speaker Verification System

Speaker verification (SV) is the process of confirming an individual's identity based on their voice characteristics. This technology is pivotal in applications ranging from security systems and forensic analysis to user authentication in various devices and services. Over the years, SV systems have witnessed significant transformations, evolving from traditional statistical models to sophisticated deep learning frameworks. This review traces the journey of SV systems, highlighting the key developments and technological advancements that have shaped the field.

In the early days of speaker verification, systems primarily relied on statistical models such as Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). These models were foundational in the development of SV systems and were widely adopted due to their ability to capture the statistical properties of speech signals.GMMs were extensively used for text-independent speaker verification. They modeled the probability distribution of speech features, allowing the system to distinguish between different speakers based on their unique vocal characteristics. The effectiveness of GMMs in handling variability within a speaker's voice made them a popular choice (Reynolds & Rose, 1995)Reynolds and Rose (1995).HMMs were particularly useful for text-dependent speaker verification, where the temporal sequence of speech was important. HMMs could model the transitions between different states in a speech signal, providing a robust framework for capturing the dynamic aspects of speech (Rabiner & Juang, 1993)Rabiner and Juang (1993). A typical traditional approach to solve the Speaker Verification problem is shown in Figure 2.1 where SVM model is used to binary classify the target.

Despite their widespread use, traditional SV methods faced several limitations:

1. **Scalability** As the size of datasets grew, GMMs and HMMs struggled to scale effectively. The computational complexity of these models made them less suitable for handling large-scale data.

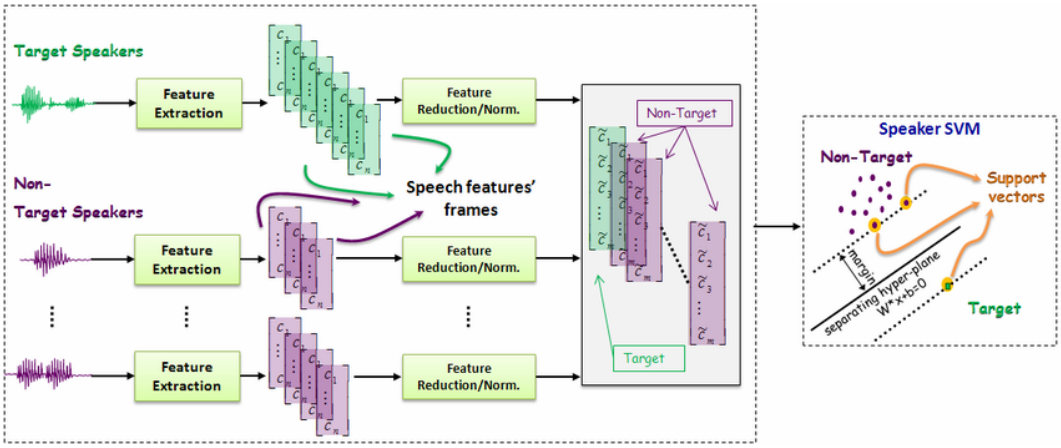


Figure 2.1: Traditional SV systems.
Fazel and Chakrabartty (2011)

2. **Complex Patterns** Traditional models had difficulty capturing complex patterns in speech signals, leading to suboptimal performance in diverse real-world scenarios.

3. **Spoofing Attacks** Traditional SV systems were vulnerable to spoofing attacks, where malicious actors could deceive the system using replayed or synthesized voices. This highlighted the need for more robust and secure verification mechanisms.

The limitations of traditional methods paved the way for the adoption of deep learning in speaker verification. Deep Neural Networks (DNNs) introduced a paradigm shift, offering a more robust and scalable framework for SV systems.The initial integration of deep learning into SV systems focused on improving feature extraction and classification. DNNs demonstrated superior performance in capturing intricate features from raw audio data, significantly enhancing the accuracy and robustness of SV systems.

Deep Speaker Embeddings: Techniques such as d-vectors and x-vectors emerged as powerful tools for speaker verification. These embeddings, generated using neural networks, captured speaker-specific characteristics more effectively than traditional methods. For example, x-vectors used time-delay neural networks (TDNNs) to extract fixed-dimensional embeddings from variable-length utterances, providing a robust representation of the speaker's voice (Snyder et al., 2018)Snyder *et al.* (2018).

End-to-End Models: The development of end-to-end deep learning models streamlined the SV process by eliminating the need for separate feature extraction and classification stages. These models learned directly from the input data, optimizing the entire pipeline with a single loss function. This approach improved accuracy and re-

duced complexity by simultaneously learning feature extraction and classification tasks (Variani et al., 2014)Variani *et al.* (2014). As deep learning techniques continued to evolve, more advanced neural network architectures were developed to further enhance SV systems. A brief overview of the approaches existing can be seen in the Figure 2.2 where Stage wise and End to End existing solutions are categorised which are discussed in 1.

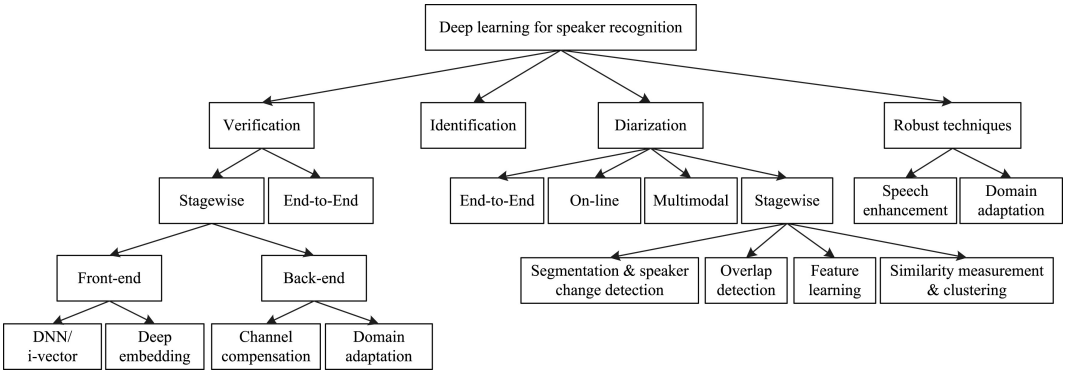


Figure 2.2: Deep NN SV systems.
Bai and Zhang (2021)

1. **Residual Networks (ResNets):** ResNets addressed the vanishing gradient problem, which often hindered the training of deep networks. By learning residual functions with reference to the layer inputs, ResNets enabled the construction of very deep networks that maintained high performance. This innovation proved beneficial in extracting robust features for speaker verification (He et al., 2016)He *et al.* (2016).

2. **Generative Models:** Generative models, including Generative Adversarial Networks (GANs), played a crucial role in improving the robustness and generalization of SV systems. GANs were used to generate synthetic data, augmenting training datasets and enhancing the model's ability to generalize to unseen conditions. This was particularly useful in scenarios where labeled data was scarce (Chou et al., 2019)Chou *et al.* (2019).

3. **Meta-Learning Strategies:** Meta-learning strategies such as prototypical networks and Siamese networks focused on learning from minimal data. These models were effective in few-shot learning tasks, where obtaining large labeled datasets was challenging. Prototypical networks, for instance, classified samples based on their proximity to prototype representations of each class, making them suitable for speaker verification tasks (Snell et al., 2017)Snell *et al.* (2017).

Several specific deep learning architectures have been developed to address the unique challenges of speaker verification.

1. **SincNet:** SincNet processes raw audio waveforms using parameterized sinc functions for convolution. This model directly learns the filter banks during training, capturing relevant speaker features more effectively than traditional methods (Ravanelli & Bengio, 2018)Ravanelli and Bengio (2018).

2. **RawNet:** RawNet operates directly on raw audio waveforms, using a series of convolutional layers and residual blocks. By processing the waveform without intermediate feature extraction steps, RawNet retains all information present in the raw audio, proving effective in text-independent speaker verification tasks (Jung et al., 2019)Jung *et al.* (2019).

3. **AM-MobileNet:** AM-MobileNet, an adaptation of MobileNet, incorporates attention mechanisms to focus on the most relevant parts of the input signal. Designed for deployment on mobile and edge devices, AM-MobileNet offers an efficient yet effective solution for speaker verification (Heo et al., 2020)Heo *et al.* (2020).

Despite the advancements brought by deep learning, several challenges remain in the field of speaker verification.

1. **Data Scarcity and Quality:** The scarcity of labeled training data, especially in diverse real-world conditions, remains a significant challenge. Techniques such as GANs for data augmentation show promise in addressing this issue by generating synthetic data to supplement training datasets (Chou et al., 2019)Chou *et al.* (2019).

2. **Robustness to Variability:** SV systems must handle various types of variability, including different recording environments, background noise, and changes in the speaker's emotional state. Domain adaptation techniques and denoising autoencoders are being explored to improve robustness under these conditions (Sun et al., 2017)Sun *et al.* (2017).

3. **Real-Time Processing:** The need for real-time processing in applications like mobile devices and security systems imposes constraints on model complexity and computational requirements. Efficient models such as AM-MobileNet address these constraints, providing accurate yet computationally efficient solutions (Heo et al., 2020)Heo *et al.* (2020).

4. **Adversarial Attacks:** SV systems are vulnerable to adversarial attacks, where maliciously crafted inputs can deceive the system. Ongoing research aims to develop techniques to detect and mitigate such attacks, enhancing the security and reliability of SV systems (Kreuk et al., 2018)Kreuk *et al.* (2018).

The evolution of speaker verification systems from traditional statistical models to advanced deep learning frameworks has significantly enhanced their performance, robustness, and scalability. While traditional methods laid the foundation, deep learning has revolutionized the field, offering sophisticated solutions to longstanding challenges. However, ongoing research is essential to address emerging issues such as data scarcity, robustness to variability, real-time processing, and adversarial attacks. As the technology continues to evolve, speaker verification systems are poised to become even more reliable and widely applicable in diverse real-world scenarios.

2.1.1 Stability v/s Plasticity Dilemma and Catastrophic Forgetting

The stability-plasticity dilemma is a fundamental challenge in the field of Neural Networks, especially relevant to neural networks and AI systems. It revolves around finding the right balance between retaining previously learned information (stability) and incorporating new information (plasticity). This section delves into the intricacies of this dilemma and the associated problem of catastrophic forgetting.

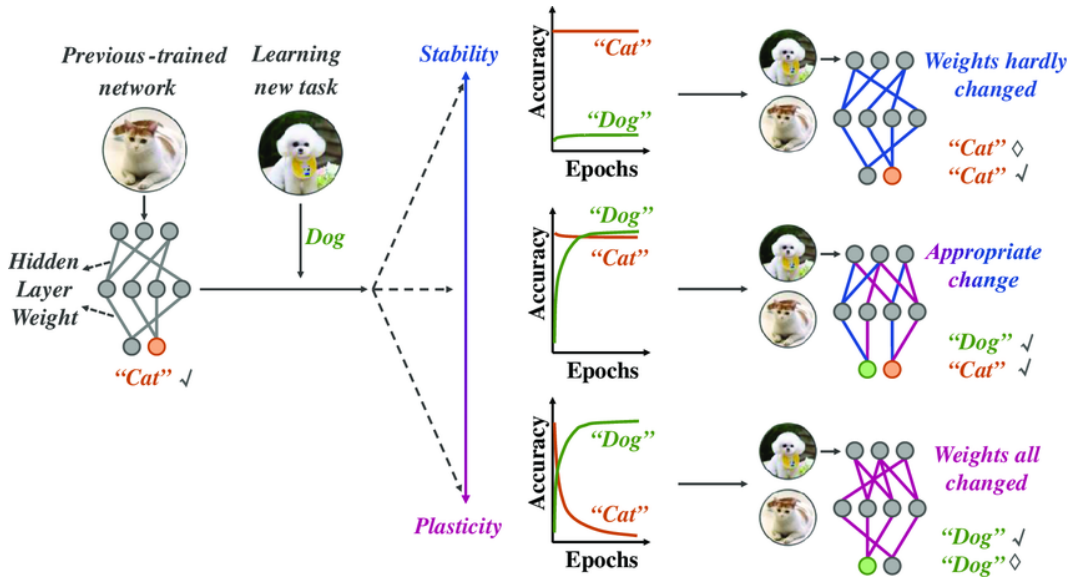


Figure 2.3: Stability v/s Plasticity Dilemma
Li *et al.* (2022)

Stability: Stability in neural networks refers to the ability to retain knowledge over time without being easily disrupted by new data. This is crucial for applications where long-term retention of learned information is necessary. For instance, in a speaker verification system, it is important to remember the characteristics of previously enrolled speakers even as new speakers are added to the system. Without sufficient stability, the system might forget old speakers, leading to verification errors.Pelosin (2023)

Plasticity: Plasticity, on the other hand, is the capacity of a neural network to adapt and learn from new experiences. This is essential for continual learning, where the system is exposed to new data and must update its knowledge base accordingly. High plasticity allows the system to integrate new information quickly and efficiently. In speaker verification, this would mean the system can effectively learn and verify new speakers as they are added. Pelosin (2023)

As shown in Figure 2.3, the Neural Network is trained over a class "Cat". When a new class "Dog" is introduced, the network retaining the Stability and structurability keeps

the previously learned weights and hence on maintaining the roubstness, didn't learn new properties of the new class and incorrectly identifies the "Dog" as "Cat". On the other hand, the network allowing the plasticity and flexibility learns all the information of the new class and forgets the previously learned properties, now incorrectly identifying the old class "Cat" as "Dog". Thus, there is always a trade off between the two "Stability v/s Plasticty" and a balance between two must be established to retain all the properties of previous as well as upcoming new classes. This issue of forgetting the previous information is known as Catastrophic Forgetting.

Catastrophic Forgetting: Catastrophic forgetting is a significant issue in continual learning where a neural network abruptly loses previously acquired knowledge upon learning new information. This happens because the network weights are updated to accommodate the new data, often at the expense of the old data. In speaker verification, this would manifest as the system losing its ability to correctly verify previously enrolled speakers after being trained on new speakers.

2.2 Continual Learning

Continual learning (CL) stands out for its unique features like explicit knowledge retention and the ability to accumulate and utilize past knowledge for future learning. However, these characteristics are not exclusive to CL. Several machine learning paradigms share related traits and objectives to varying extents. This section explores the most closely related paradigms—transfer learning, multi-task learning, online learning, reinforcement learning, meta-learning, curriculum learning, and sequential learning—and distinguishes them from continual learning.

2.2.1 Related Learning Paradigms

Transfer Learning

Transfer learning is a prominent area in machine learning and deep learning, often referred to as domain adaptation in fields like computer vision and natural language processing. It typically involves two domains: a source domain with ample labeled training

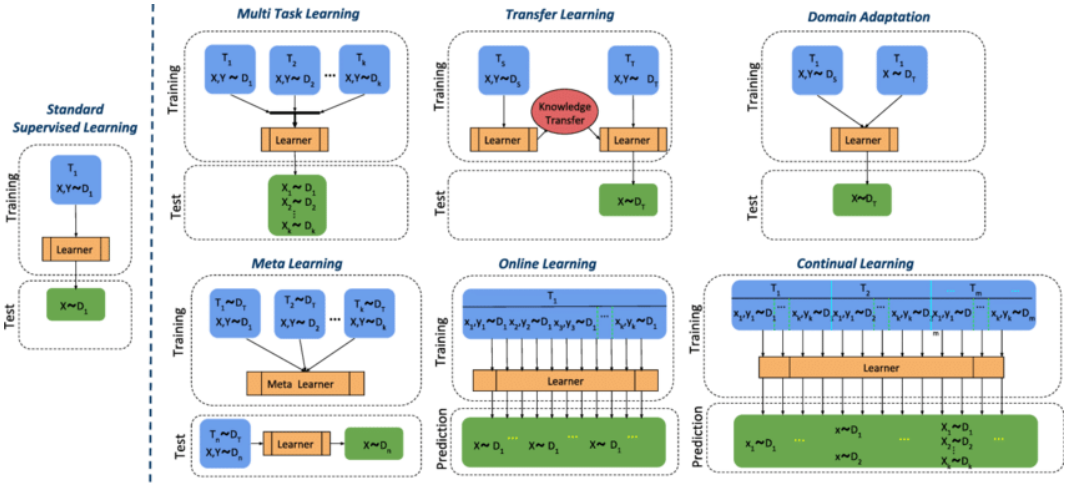


Figure 2.4: Types of Learning Paradigms
De Lange *et al.* (2022)

data and a target domain with little or no labeled data. The goal is to leverage the labeled data from the source domain to enhance learning in the target domain Long *et al.* (2017) Pan (2020) Taylor and Stone (2009).The primary difference between transfer learning and continual learning is that transfer learning does not focus on continuous learning or knowledge accumulation. As we can see in Figure 2.4 the transfer of knowledge occurs in a single step, and the ability to solve the task in the source domain is generally not retained. Continual learning, on the other hand, emphasizes the retention and accumulation of knowledge, enabling systems to become progressively more knowledgeable and facilitating easier learning of new tasks. Furthermore, transfer learning is typically unidirectional, transferring knowledge from the source to the target domain, whereas continual learning allows bidirectional knowledge transfer, where learning from new domains can enhance previous domains.

Multi-Task Learning

Multi-task learning involves learning multiple related tasks simultaneously to improve performance by sharing relevant information across tasks Caruana (1997). The idea is to introduce an inductive bias into the joint hypothesis space of all tasks, which helps prevent overfitting and enhances generalization. While multi-task learning shares the goal of utilizing shared information across tasks with continual learning, it operates within the traditional learning paradigm. Multi-task learning optimizes several tasks at once, effectively treating them as one larger task. This simultaneous optimization

requires retaining all encountered training data, which is impractical for long-term scalability. In contrast, continual learning focuses on learning tasks sequentially, retaining and building on past knowledge to support future learning.

Meta-Learning

Meta-learning, or "learning to learn," uses metadata about past experiences to enhance learning on new tasksAndrychowicz *et al.* (2016). As shown in Figure 2.4 while it shares the continual learning objective of ongoing learning, most meta-learning research focuses on accelerating learning on fixed datasets or specific target domains without revisiting the source domain. Meta-learning typically involves a dual-system approach: one system for actual learning and another to guide the learning process. Continual learning, however, emphasizes the continuous accumulation and application of knowledge across a sequence of tasks, without the strict dual-system structure.

Online Learning

Online learning processes training data one example at a time, updating the model incrementally with each new data point. This approach is used when it is computationally infeasible to train over the entire dataset or when mini-batch learning is impractical due to hardware constraints or data availability. Online learning is efficient in terms of memory and runtime, which is crucial in real-world scenarios Kivinen *et al.* (2004) Mairal *et al.* (2010).Although online learning involves streaming data, its objective is different from that of continual learning. Online learning aims to optimize performance on a single, ongoing task, whereas continual learning focuses on learning from a sequence of different tasks, retaining knowledge, and using it to improve future learning. Dredze *et al.* (2008)

Reinforcement Learning

Reinforcement learning (RL) involves an agent learning through trial and error interactions with a dynamic environment. The agent receives inputs about the current state and potential rewards, selects actions, and learns optimal policies that maximize future rewards Sutton (1999) Van Otterlo and Wiering (2012). Although RL seems inherently

continual, most RL research focuses on a single task within a stationary environment. Transfer and multi-task learning can be applied to RL to speed up learning and improve policy optimization, but RL generally does not emphasize knowledge retention and accumulation for future tasks as continual learning doesBarrett *et al.* (2010) Taylor and Stone (2009).

Curriculum Learning

Curriculum learning involves presenting tasks or data to a learning algorithm in a sequence that facilitates learning of increasingly complex tasks. Both curriculum learning and continual learning handle sequences of tasks, but curriculum learning arranges tasks to progressively ease learning of the final task. In contrast, continual learning aims to retain the ability to solve all encountered tasks, not just the final oneBengio *et al.* (2009).

Sequential Learning

Sequential learning, often referred to as time-series forecasting or predictive learning, deals with ordered observations that must be preserved during model training and prediction. Unlike other supervised learning problems, sequential learning emphasizes the temporal order of data, making it distinct from other learning paradigmsSutskever *et al.* (2014).

Continual Learning in AI

The concept of continual learning has emerged prominently in AI, often referred to as Lifelong Learning or Continuous Learning. While the term "Lifelong Learning" has been used for years, it is now increasingly associated with deep learning algorithmsParisi and Lomonaco (2020) Liu (2020). "Continual Learning" conveys the idea of frequent intervals of learning, unlike "Continuous Learning," which implies uninterrupted learning and might be confusing in some contexts, especially in reinforcement learning.

Terminological Clarifications

"Online learning" contrasts with "Batch Learning" by processing data sequentially rather

than all at onceCui *et al.* (2016). "Incremental Learning," though focused on building knowledge incrementally, does not necessarily imply the adaptability inherent in continual learning, where previously learned knowledge might need to be adjusted or even forgotten to accommodate new information.

In summary, while several machine learning paradigms share characteristics with continual learning, they differ in their objectives and operational methods. Continual learning uniquely emphasizes the ongoing accumulation and retention of knowledge, enabling systems to learn and adapt continually over time.

2.2.2 Continual Learning Framework

The concept of continual learning (CL) is pivotal for developing truly intelligent AI systems. Bing Liu emphasized that without lifelong learning capabilities, AI systems cannot achieve true intelligence. Continual learning involves a computer program that improves its performance in approximating a target function h^* through sequential experiences E_i, each being transient and partial.

Formal Definition

CL tackles Probably Approximately Correct (PAC) learnable problems by approximating a target hypothesis h^* from a sequence of non-i.i.d. training batches. The framework builds on previous models, assuming learning occurs from a potentially infinite sequence of distributions $D = \{D_1, ..., D_n\}$ where D_i represents Data at current task T.

Definition

Given X and Y as input and output random variables, consider D a potentially infinite sequence of unknown distributions $D = \{D_1, ..., D_n\}$ over $X \times Y$, encountered over time. A continual learning algorithm A_{CL} has the following signature:

$$\forall D_i \in D, A^i_{CL} :< h_{i-1}, B_i, M_{i-1}, t_i > \rightarrow < h_i, M_i >$$

, where M_i is an external memory storing previous training examples or partial computations not directly related to the model's parameters,t_i is a task label, void if not provided. It can disentangle tasks and specialize hypothesis parameters. B_i is the training batch of examples, assumed to be drawn i.i.d. from D_i but not necessarily. Examples can also be drawn non-i.i.d. Each D_i can be considered a stationary distribution. Each B_i consists of examples e_{ij} with $j \in [1, ..., |B_i|]$. Each example $e_{ij} =< x_{ij}, f_{ij} >$, where f_i is the feedback signal used to infer the optimal hypothesis $h^*(x, t)$.

The objective of a CL algorithm is to minimize the loss $\mathcal{L}_S$ over the entire stream of data S:

$$\mathcal{L}_S(f_n^{CL}, n) = \frac{1}{\sum_{i=1}^{n} |\mathcal{D}_{\text{test}}^i|} \sum_{i=1}^{n} \mathcal{L}_{\text{exp}}(f_n^{CL}, \mathcal{D}_{\text{test}}^i) \tag{2.1}$$

$$\mathcal{L}_{\text{exp}}(f^{CL}, \mathcal{D}_{\text{test}}^i) = \sum_{j=1}^{|\mathcal{D}_{\text{test}}^i|} \mathcal{L}(f_n^{CL}(x_j^i), y_j^i) \tag{2.2}$$

, where the loss $\mathcal{L}(f_n^{CL}(x), y)$ is computed on a single sample $\langle x, y \rangle$, such as cross-entropy in classification problems. Here, the respective variables have meaning as:
$\mathcal{L}_S$: The overall loss over the entire stream of data S.
S: The entire stream of data.
f_n^{CL}: The continual learning model at step n.
n: The number of steps or tasks in the continual learning process.
$\mathcal{L}$exp($f^{CL}n, \mathcal{D}^i$test): The loss experienced on the test dataset $\mathcal{D}^i$test for the i-th task.
$\mathcal{D}_{\text{test}}^i$: The test dataset for the i-th task.
$|\mathcal{D}^i$test$|$: The number of samples in the test dataset $\mathcal{D}^i$test.
$\mathcal{L}(f_n^{CL}(x_j^i), y_j^i)$: The loss for a single sample j from the i-th test dataset, computed using the model f_n^{CL}.
x_j^i: The j-th input sample from the i-th test dataset. y_j^i: The corresponding label for the j-th input sample from the i-th test dataset.
In this context, the loss $\mathcal{L}(f_n^{CL}(x), y)$ is typically the cross-entropy loss used in classification problems. The overall objective is to minimize the cumulative loss across all tasks in the continual learning processLomonaco (2019).

Task Notation A task T is defined by a unique task label $\hat{t}$ and its target function

$g_{\hat{t}}^*(x) \equiv h^*(x, t = \hat{t})$, the objective of its learning. If t is not given as input to the CL algorithm A_{CL}, with all $t_i = \emptyset$, it is like having a single incremental task T where $g_t^* \equiv h^*$. Disentangling the notion of task from the training batch is crucial in CL since data are not available all at once. They may be related to the same learning objective g_t^* defined by the external supervised signal t. Hence, even if D_i represents a different distribution from D_j for $i \neq j$, this does not necessarily define a different task. Lomonaco (2019)

Constraints, Relaxations, and Desiderata

Constraints

1. **External Memory:** The external memory M should be significantly smaller than the total number of training examples encountered.
2. **Memory and Computation:** Memory usage and computational complexity per iteration are bounded by reasonably small values.

Relaxations

1. **Memory Relaxation:** Removes fixed memory bounds.
2. **Computation Relaxation:** Removes fixed computational bounds.

Desiderata

1. **Storage-Free Continual Learning:** Avoids using external memory.
2. **Online Continual Learning:** Processes data in real-time of batch size $|B_i| = 1$.

Scenarios

Three common CL scenarios are defined based on task-awareness and task-agnosticism:

1. **Multi-Task (MT):** Each task has a distinct task label.
2. **Single-Incremental-Task (SIT):** A single, incremental task without distinct task labels.
3. **Multiple-Incremental-Task (MIT):** Tasks with overlapping distributions, requiring task-specific behaviors and generalization.

Update Content Types

1. **New Instances (NI):** New examples of previously encountered classes.
2. **New Classes (NC):** Examples from new, previously unseen classes.
3. **New Instances and Classes (NIC):** A mix of new examples from old and new classes.

This formal framework for CL, with its constraints, relaxations, and scenarios, aims to provide a structured foundation for developing and assessing continual learning algorithms. This approach enhances the potential for creating more adaptable and intelligent AI systemsLomonaco (2019).

2.2.3 CL Techniques and Strategies

Given the complexity of various real-world applications, continual learning manifests in diverse scenarios across different settings. In this thesis, we categorize continual learning scenarios in speech and audio tasks into three primary types: data-incremental, class-incremental, and task-incremental learning.

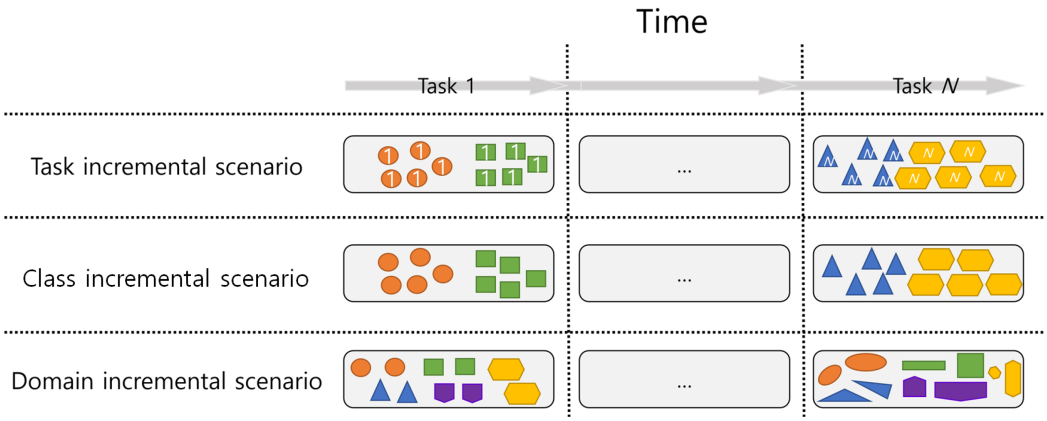


Figure 2.5: Scnarios in Continual Learning
Ha *et al.* (2023)

Data-incremental learning

It involves a setting where there are no explicit task boundaries, and the model continually learns from an incremental data streamVan de Ven *et al.* (2022). Sequence modeling tasks, such as automatic speech recognition, are examples of this setting. Here, the model continuously trains on data streams to solve the same or similar problems within a shared vocabulary label space. Because the model lacks explicit task boundaries, it

must either process data online or infer task identifiers implicitly from the current data stream's statistics.

Class-incremental learning

It refers to scenarios where new, unseen classes appear continuously over time Liu *et al.* (2023) as we can see in Figure 2.5. Utterance classification tasks like acoustic scene classification and intent classification exemplify this scenario. Each task may have a unique label space, a subset of the entire label space Y. The objective is to develop a comprehensive classifier that can manage all encountered classes from past data.

Task-incremental learning

It is more general, covering scenarios where the model incrementally acquires new capabilities ass shown in Figure 2.5 by expanding to handle more tasks using knowledge transfer from previous tasks Hossain *et al.* (2022). This can apply to tasks such as intent classification and slot filling. Regression tasks like speech enhancement also fit this category, where the model adapts to changes in data distribution, such as different noise types, enhancing overall robustness.

Continual Learning Methods

To address the various continual learning scenarios in real applications, several types of methods have been proposed. These methods can be grouped into three categories based on the module they operate on within the general ML pipeline: rehearsal, architecture, and regularization-based methods, as depicted in Figure 2.6.

1. **Rehearsal methods:** It retain examples from past data in a memory buffer and replay them when learning new tasks to mitigate catastrophic forgetting Yang (2023). Some methods store raw samples or their representations, while others use generative models to produce pseudo data. These methods typically perform well but require memory storage M. For each task t, samples from previous task M_{t-1} and the current data stream D_t are combined to form the training data. This helps reduce the forgetting effect by recalling past samples during training.
2. **Architecture-based methods:** It expand model parameters by isolating parameters for different tasks Yang (2023). The aim is to minimize interference between

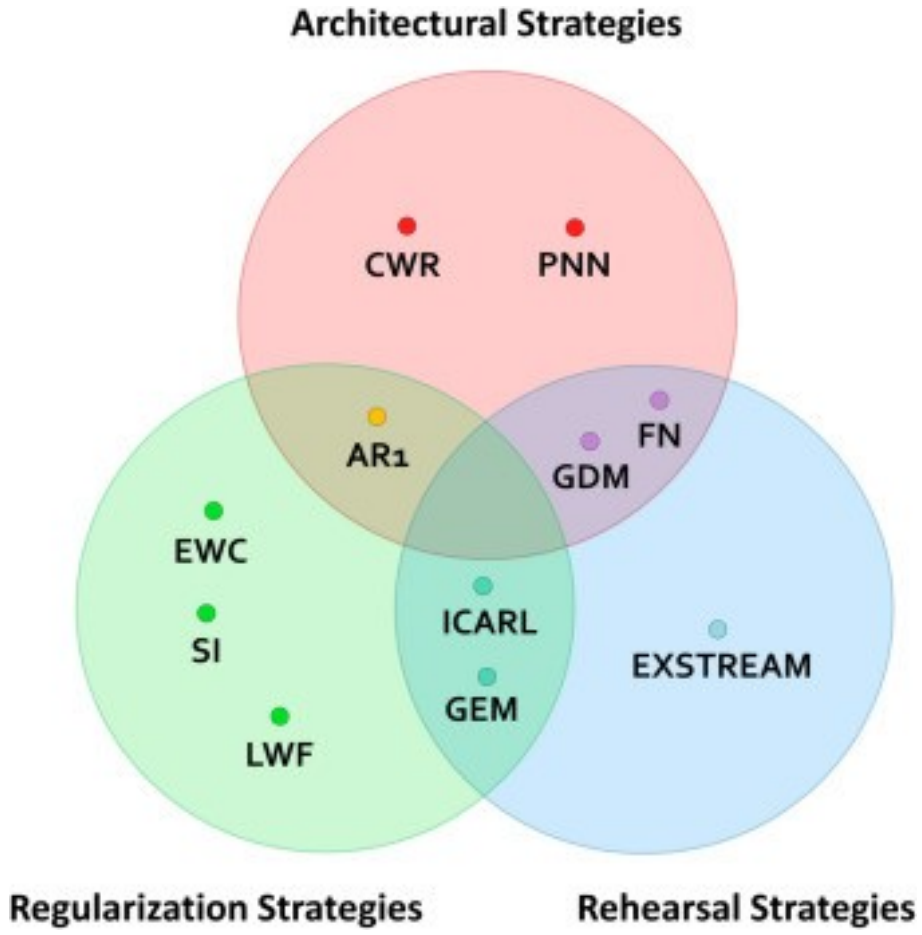


Figure 2.6: Strategies in Continual Learning
Lomonaco (2019)

different model modules. Each task t has task-specific and task-sharing parameters. Task-sharing parameters are updated across all tasks, while task-specific parameters are updated only for the current task and frozen otherwise. These methods usually require a task oracle to activate corresponding task branches during inference.

3. **Regularization methods:** Regularization methods add a regularization term when encountering new tasks to adjust parameter importance, thereby improving continual learning at the loss level Yang (2023). The goal is to consolidate the learned knowledge in the model, preventing forgetting. The importance of model parameters is estimated to ensure feasibility. The loss function is updated as follows:

$$\ell(\theta_t) \leftarrow \ell(\theta_t) + \frac{\lambda}{2}(\theta_t - \theta_{t-1})^\top W_{t-1}(\theta_t - \theta_{t-1}) \tag{2.3}$$

where λ is the regularization coefficient controlling the weight of regularization. Definitions of Variables with Proper Notations can be understood below: $\ell(\theta_t)$: The loss function at time step t, θ_t: The model parameters at time step t, θ_{t-1}: The model parameters at the previous time step $(t-1)$, λ: The regularization coefficient that controls the weight of the regularization term, W_{t-1}: A matrix representing the importance of model parameters estimated from the previous task at time step $(t-1)$ and $(\theta_t - \theta_{t-1})^\top$: The transpose of the difference between the model parameters at time steps t and $(t-1)$. The regularization term

$\frac{\lambda}{2}(\theta_t - \theta_{t-1})^\top W_{t-1}(\theta_t - \theta_{t-1})$ is added to the original loss function $\ell(\theta_t)$ to help consolidate the learned knowledge and prevent forgetting by adjusting the importance of parameters when encountering new tasks.

2.3 Application of Continual Learning in Speaker Verification

Case Study: Consider a corporate environment where an SV system is deployed to secure access to sensitive areas. New employees are frequently added, and existing employees may change their speech patterns due to various factors. A CL-based SV system can adapt to these changes by continuously updating speaker models. When a new employee is added, the system incrementally learns their voice characteristics, ensuring accurate verification without affecting the performance for existing employees. Similarly, as employees' voices change, the system adapts to maintain high verification accuracy.

2.3.1 Advantages, Challenges and Implementation

Continual learning (CL) has the potential to significantly enhance speaker verification (SV) systems by enabling them to adapt to new speakers and changing environments over time. This section explores the application of CL in SV, highlighting its benefits, challenges, and implementation strategies.

Adaptive Learning: CL allows SV systems to continuously update and refine speaker models as new speech data becomes available. This capability is especially useful in real-world applications where new users are frequently added, or existing users provide additional speech samples over time. By leveraging CL, SV systems can maintain high verification accuracy in dynamic settings without requiring complete retraining.

Personalization: CL enables SV systems to personalize models for individual users. As users interact with the system and provide more data, CL algorithms incrementally update speaker models to better reflect each user's unique voice characteristics. This personalization enhances system accuracy and user experience, making it more responsive and reliable.

Robustness to Variability: Speech signals can vary due to aging, health changes, emotional states, and different recording conditions. CL helps SV systems adapt to these variations, ensuring consistent performance. For instance, as a user's voice changes due to aging, CL algorithms update the speaker model incrementally to maintain verification accuracy.

Implementation Strategies:

1. **Semi-Supervised Learning:** CL can use semi-supervised learning strategies, updating models with both labeled and unlabeled data. This reduces the need for extensive labeled datasets and allows the system to learn from real-world usage scenarios.
2. **Incremental Model Updates:** Incremental updates involve periodically updating speaker models with new data while retaining knowledge of previously seen data. Techniques like fine-tuning adjust model parameters using new speech samples without forgetting old ones.
3. **Dynamic Memory Networks:** These networks manage and update speaker models efficiently using a memory component to store and retrieve speaker-specific information dynamically, allowing the system to adapt to new data while preserving past knowledge.
4. **Multi-Task Learning:** Multi-task learning frameworks can be adapted for CL in SV, where the system simultaneously learns multiple tasks such as speaker verification, identification, and diarization. Sharing knowledge across tasks improves generalization and robustness to new data.
5. **Reinforcement Learning:** Reinforcement learning can be applied to CL in SV, with the system learning to update speaker models based on feedback from verification performance. The RL agent maximizes verification accuracy by incrementally adjusting model parameters as new data is encountered.

Despite the benefits, CL in SV faces challenges such as efficient memory management, handling noisy and imbalanced data, and ensuring real-time updates without compromising performance. Future research can explore advanced CL techniques, like meta-learning and generative replay, to address these challenges and enhance the applicability of CL in SV. The application of continual learning in speaker verification has the potential to create more adaptive, personalized, and robust systems. By leveraging CL techniques, SV systems can maintain high accuracy and reliability in dynamic and real-world environments, addressing the evolving needs of users and applications Yang *et al.* (2022) Fu *et al.* (2021).

CHAPTER 3

Proposed Work

3.1 Problem Statement

Speaker Verification thus need a robust solution which can incorporate the upcoming new stream of data with retaining the Plasticity in the network. We again state our Problem Statement to provide deeper insights into the problem as:
"This thesis delves into the realm of continual learning for large-scale speaker verification systems, aiming to optimize strategies that effectively address the challenges posed by evolving speaker datasets, mitigate the risk of catastrophic forgetting, and foster adaptability for sustained performance excellence in dynamic environments."

To implement this we surveyed over a lot of datasets available for Speaker Verification. Many datasets are telephony and clean speech datasets which contain only single language or single speaker. We focused on working on such a dataset which is very well diverse and has real world challenges. A list of existing datasets can be seen in Table 3.1.

Table 3.1: Available Datasets for Speaker Verification

Name	**Cond.**	**Free**	**No of Speakers**	**No of Utter.**
ELSDSR (Feng and Hansen, 2005)	Clean Speech	Yes	22	198
MIT Mobile (Woo et al., 2006)	Mobile Devices	No	88	7884
SWB (Godfrey et al., 1992)	Telephony	No	3114	33,039
POLYCOST (Hennebert et al., 2000)	Telephony	No	133	1285‡
ICSI Meeting Corpus (Janin et al., 2003)	Meetings	No	53	922
Forensic Comparison (Morrison et al., 2015)	Telephony	Yes	552	1264
ANDOSL (Millar et al., 1994)	Clean speech	No	204	33,900
TIMIT (Fisher et al., 1986)†	Clean speech	No	630	6300
MGB Challenge Dataset (Bell et al., 2015)	Broadcast Data	**	Unknown	1600 hs
SITW (McLaren et al., 2016)	Multi-media	Yes	299	2800
NIST SRE Greenberg (2012)	Clean speech	No	2000+	*
VoxCeleb1	Multi-media	Yes	1251	153,516
VoxCeleb2	Multi-media	Yes	6112	1,128,246

We aimed to apply Continual learning framework as a solution to solve this Large Scale Speaker Verification problem on the Voxceleb Dataset. We narrowed down our approach to the class incremental scenario assuming the classes are increasing with time.

There are 2 types of CL scenarios which includes Online Continual learning and Offline Continual learning.

1. **Online CL:** In this type of Continual learning scenario data comes in real time. Ex - A robot picking a ball. Everytime he picks up the ball a real time data is generated from it.
2. **Offline CL:** In this scenario the data is pre available to us, the only difference is the dataset is moulded into stream of data (defined as Task in earlier section) and is feeded to the model as real time scenario.

We selected Voxceleb2 as our dataset to implement this Class incremental strategy and worked on Offline CL scenario to solve the problem. Voxceleb2 itself is 1 TB in size which makes this problem a really challenging task as retraining itself over the entire dataset takes too much time posing time and complexity issues. To handle this situation we created a Data split strategy which will be discussed in further sections.

3.2 Datasets

In this thesis, our proposed methods are evaluated on a text-independent speaker verification (SV) task conducted under noisy and unconstrained conditions. This task is chosen for two primary reasons: First, speaker verification in such challenging environments is not only difficult but also highly relevant for practical applications, as methods developed can be directly applied to real-world scenarios. Second, the datasets used for this task contain millions of utterances from thousands of speakers as we can see in Figure 3.1, all of which are publicly available, making the research reproducible and verifiable.

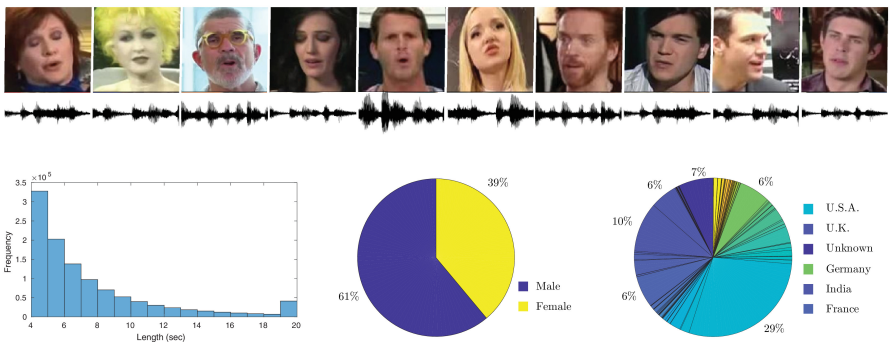


Figure 3.1: Voxceleb2 Dataset preview
Nagrani *et al.* (2017)

3.2.1 Voxceleb2

The primary training dataset used in our system is VoxCeleb2. VoxCeleb2 consists of over 1 million utterances from more than 6000 celebrities, sourced from YouTube videos. The dataset is balanced in terms of gender and showcases a wide variety of accents, ages, professions, and ethnicities. The audio recordings come from diverse and challenging acoustic environments, including Celebrity interviews on red carpets, Speeches given in outdoor stadiums and indoor studios, Excerpts from professionally shot multimedia and Videos recorded with handheld devices.

Table 3.2: Dataset statistics for VoxCeleb 1&2

Dataset	**VoxCeleb1**	**VoxCeleb2**
No of speakers	1251	6112
No of male speakers	690	3761
No of videos	22,496	150,480
No of hours	352	2442
No of utterances	153,516	1,128,246
Avg No of videos per speaker	18	25
Avg No of utterances per speaker	116	185
Avg length of utterances (s)	8.2	7.8

All speech data in VoxCeleb2 are subject to various types of noise, such as background chatter, laughter, overlapping speech, and channel noise from different recording equipment. Table 3.2 shows the general statistics of the VoxCeleb dataset. The evaluation of the systems is performed using VoxCeleb1 dataset as a test set for Voxceleb2 dataset.VoxCeleb1 is collected using a similar pipeline to VoxCeleb2 but is smaller in size. It is roughly gender-balanced with 55% male speakers, and the speech segments are also subject to various real-world noises. For evaluation purposes, VoxCeleb1 can be split into three subsets:

VoxCeleb1-O: The original evaluation set of VoxCeleb1.
VoxCeleb1-E: The entire VoxCeleb1 dataset, including both the development and evaluation sets.
VoxCeleb1-H: A challenging subset of VoxCeleb1, consisting of data pairs with the same nationality and gender Nagrani *et al.* (2017).

3.3 Methodology

3.3.1 Class-Incremental Strategy

We narrowed down our approach to the class-incremental scenario. The class-incremental technique creates a stream of data as in Figure 3.2 where new classes are continuously introduced over period of time. As our implementation involve Offline CL we had the opportunity to overlook on the whole data beforehand. Provided 6000 different speakers in Voxceleb2 the problem stood pretty complex giving Computational as well as Time Complexity issues as the data streams obtained are very long. This lead us to the study of Voxceleb2 which gave an intuition of creating Data Splits effectively as per our use case.

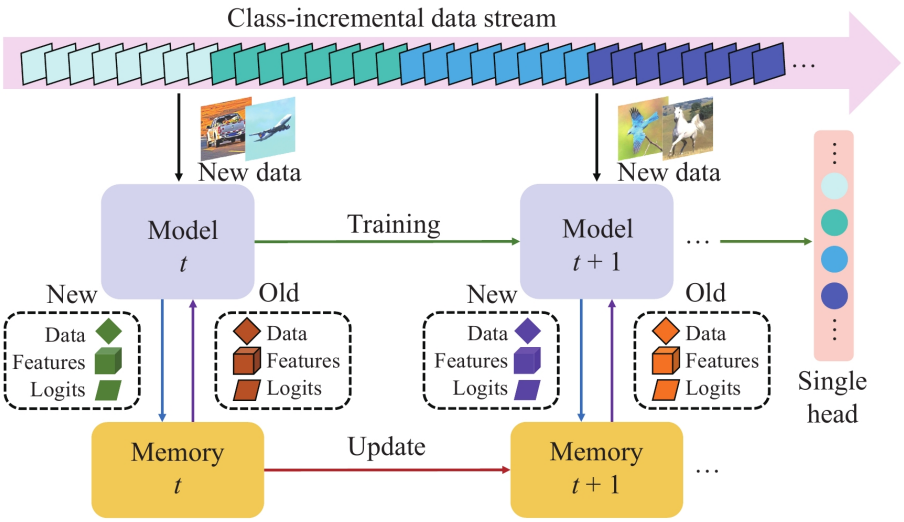


Figure 3.2: Class-Incremental Scenario
Da Yu (2023)

3.3.2 Data Splits

We created Data Splits of the Overall Data Stream to map it to the Continual learning setting and created a Base Split of 60% as in Figure 3.3 of the data including the classes having very high variance. This helped us in including those classes which may act as outliers. With new classes introducing such a class which is outlier can lead to total Data Distribution shift, thus showing the model diverse data in the 1st pass of the data stream results in better training and learning of the model. We selected 100 samples randomly of each classes and observed classes with high variance. This high variance classes are

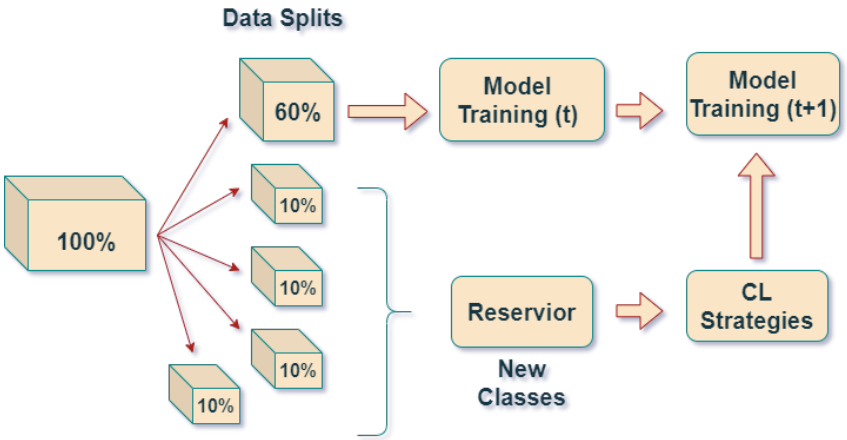


Figure 3.3: Handling New Classes

included in the 60% data split in the 1st pass. The rest of the classes are introduced with 10% split in 4 stages accounting for the rest of the 40% classes. In an Online CL scenario when new data comes in real time, we simulated the same by introducing data splits in each pass of the training of the model. This fabrication mitigates the cost of entire retrainig of the model when a totally new class is introduced in the data stream. We still faced issues of Class imbalance, Data Domain Shift, Memory Constraints & Evaluation Difficulty over period of time and we used Continual learning methods to handle the issues.

3.3.3 CL Methods

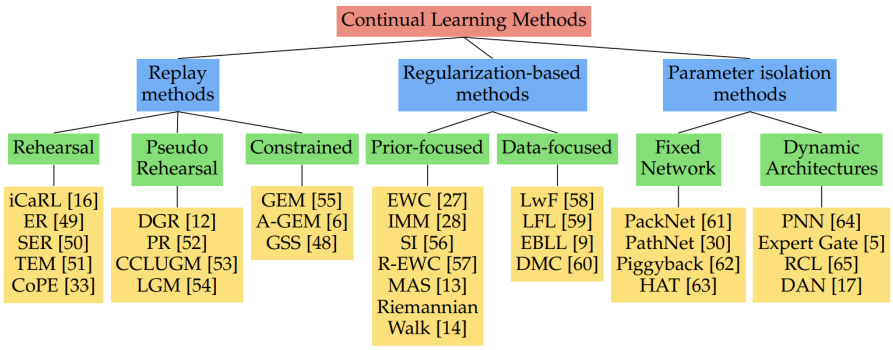


Figure 3.4: CL Methods
De Lange *et al.* (2022)

There exist many CL methods which can work effectively in our scenario. Figure 3.4 shows not exhaustive but the list of methods which can be clubbed together with different strategies and scenarios to handle issues like either class imbalance or domain shift, etc. We focused on two main methods that are widely used and very effective in

class-incremental scenario. The first one helps in retaining the stability as it focuses on penalizing weights and the other retains the plasticity as it focuses on retraining previous examples.

Elastic Weight Consolidation (EWC):

Definition: Elastic Weight Consolidation (EWC) is a regularization technique used in continual learning to prevent catastrophic forgetting. Catastrophic forgetting occurs when a neural network trained sequentially on multiple tasks forgets the previously learned tasks upon learning new onesKirkpatrick *et al.* (2017).

Mechanism: EWC mitigates this by adding a regularization term to the loss function, which penalizes changes to the weights that are crucial for previously learned tasks.

Mathematical Formulation: The modified loss function for a new task B after learning task A is:

$$L(\theta) = L_B(\theta) + \frac{\lambda}{2} \sum_i F_i(\theta_i - \theta^*_{A,i})^2$$

, where $L_B(\theta)$ is the loss for the new task, θ_i are the model parameters, $\theta^*_{A,i}$ are the optimal parameters learned for task A, F_i is the Fisher Information Matrix indicating the importance of each parameter and λ is a hyperparameter controlling the strength of the regularization. A graphical overview of EWC is shown in Figure 3.5.

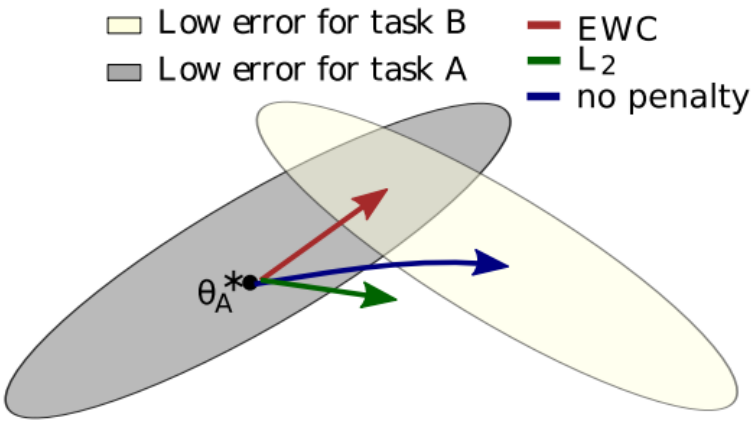


Figure 3.5: Elastic Weight Consolidation Method.
Kirkpatrick *et al.* (2017)

Replay Method:

Definition: Replay methods, including Experience Replay and Generative Replay, are techniques used in continual learning to combat forgetting by rehearsing previous experiencesPellegrini *et al.* (2019).

Mechanism: The method involves storing a subset of the training data from previous tasks and periodically replaying it while learning new tasks.

Experience Replay: Directly stores and replays samples from previous tasks.

Generative Replay: Uses a generative model to generate synthetic data resembling previous experiences.

Process: When training on a new task, the model is simultaneously trained on the new task data and the replayed old task data, thereby maintaining performance on both. A systematic working algorithm can be seen in Figure 3.6

Algorithm 1 Pseudocode explaining how the external memory RM is populated across the training batches. Note that the amount h of patterns to add progressively decreases to maintain a nearly balanced contribution from the different training batches, but no constraints are enforced to achieve a class-balancing.

$RM = \varnothing$
$RM_{size} =$ number of patterns to be stored in RM
for each training batch B_i:
  train the model on shuffled $B_i \cup RM$
  $h = \dfrac{RM_{size}}{i}$
  $R_{add} =$ random sampling h patterns from B_i
  $R_{replace} = \begin{cases} \varnothing & \text{if } i == 1 \\ \text{random sample } h \text{ patterns from } RM & \text{otherwise} \end{cases}$
  $RM = (RM - R_{replace}) \cup R_{add}$

Figure 3.6: Replay Method.
Pellegrini *et al.* (2019)

Hybrid Approach:

Concept: A hybrid approach leverages the strengths of both EWC and Replay methods to further reduce catastrophic forgetting.

Mechanism:

1. *Regularization:* EWC is applied to ensure that the model parameters important for previous tasks are preserved.

2. *Rehearsal:* Replay of old task data (either actual data or generated) is performed during the training of new tasks.

This combined strategy addresses forgetting at both the parameter level (through EWC) and the data level (through Replay), leading to more robust continual learning.

Implementation: The loss function in the hybrid approach can be formulated as:

$$L(\theta) = L_B(\theta) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta^*_{A,i})^2 + \beta \cdot L_R(\theta)$$

, where $L_R(\theta)$ is the loss from replaying old task data and β is a hyperparameter balancing the replay loss with the regular task loss and EWC regularization.

3.4 Experimental Setup

We performed many experiments and rested our case on below 5 major experiments:

1. Baseline 60% base + 40% in 4 stages with 10% each
2. Distribution based 60% high variance + 40% in 4 stages with 10% each
3. EWC 90% base + 10% each for each state dict of model at time t
4. Replay Replay 10% of data randomly in each new split from previous split
5. EWC+ Replay Hybrid of both EWC split and Replay

3.4.1 Model Architecture

We selected ECAPA-TDNN(Emphasized Channel Attention, Propagation, and Aggregation in TDNN) as our model. It is a novel architecture designed to enhance speaker verification systems. It builds on the Time Delay Neural Network (TDNN) framework and incorporates several advanced techniques inspired by recent developments in face verification and computer vision. Below is a detailed overview of the ECAPA-TDNN model based on the paper "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification" by Brecht Desplanques, Jenthe Thienpondt, and Kris DemuynckDesplanques *et al.* (2020). An overview of the model can be seen in Figure 3.7.

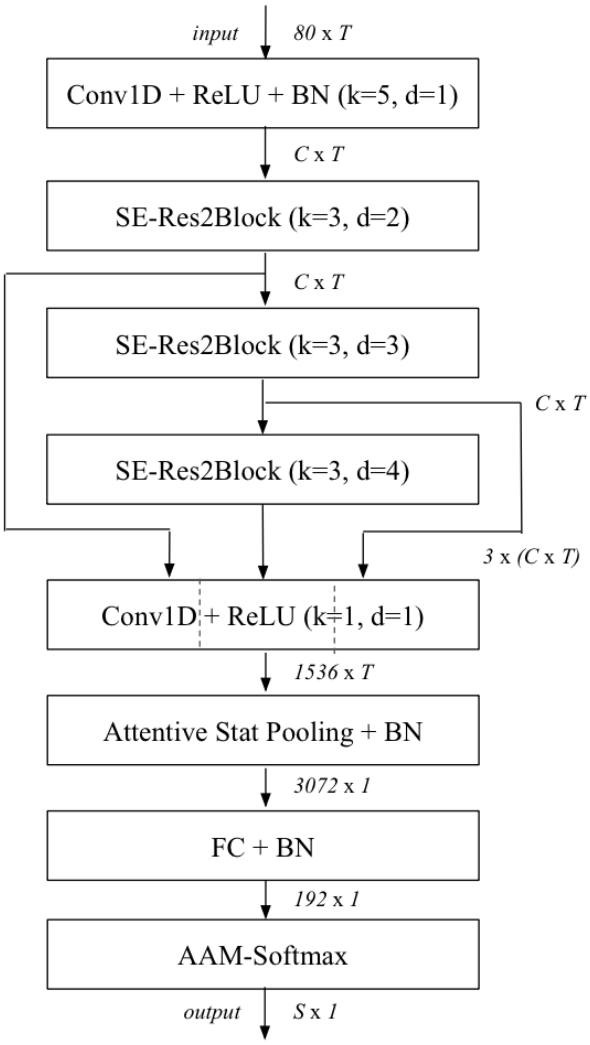


Figure 3.7: ECAPA_TDNN Architecture.
Desplanques *et al.* (2020)

Key Components and Enhancements

1. **1-Dimensional Res2Net Modules**
2. **Squeeze-and-Excitation (SE) Blocks**
3. **Channel- and Context-Dependent Statistics Pooling**
4. **Multi-Layer Feature Aggregation and Summation**
5. **SE-Res2Blocks**

3.4.2 Evaluation Metrics

3.4.3 Equal Error Rate (EER):

Definition: Equal Error Rate (EER) is the point at which the false acceptance rate (FAR) equals the false rejection rate (FRR). It is a single metric that provides a balanced measure of a system's accuracy.
Formula: EER is found at the point where:

$$\text{FAR} = \text{FRR}$$

The actual calculation involves plotting the Receiver Operating Characteristic (ROC) curve and finding the point where the FAR and FRR curves intersect. **Loss Function:** While EER itself is not derived from a loss function, it can be related to the cost of errors in a decision-making process. Typically, minimizing EER involves optimizing thresholds in the system to balance FAR and FRRCheng and Wang (2004).

3.4.4 Minimum Detection Cost Function (min DCF):

Definition: Minimum Detection Cost Function (min DCF) is a metric used to evaluate the performance of biometric verification systems by considering both the false acceptance rate (FAR) and the false rejection rate (FRR), weighted by their respective costs and prior probabilities Greenberg *et al.* (2014).
Formula: The DCF is calculated as:

$$\text{DCF} = C_{\text{miss}} \cdot P_{\text{miss}} \cdot P_{\text{target}} + C_{\text{fa}} \cdot P_{\text{fa}} \cdot (1 - P_{\text{target}})$$

where C_{miss} is the cost of a false rejection (miss), P_{miss} is the probability of a miss (FRR), P_{target} is the prior probability of the target (the probability that the person is an actual match), C_{fa} is the cost of a false acceptance and P_{fa} is the probability of a false acceptance (FAR).

Loss Function: The DCF can be interpreted as a loss function that combines both types of errors (false acceptances and false rejections) and their associated costs. Minimizing the DCF involves finding the threshold that yields the lowest combined cost of these

errors:

$$\text{min DCF} = \min_{\theta} \left(C_{\text{miss}} \cdot P_{\text{miss}}(\theta) \cdot P_{\text{target}} + C_{\text{fa}} \cdot P_{\text{fa}}(\theta) \cdot (1 - P_{\text{target}}) \right)$$

where θ represents the decision threshold. Both EER and min DCF are crucial metrics for evaluating and optimizing the performance of Speaker Verification systems, providing insights into the trade-offs between different types of errors and their costs.

3.4.5 Model Training

The ECAPA-TDNN model significantly outperforms traditional TDNN and ResNet-based speaker verification systems by incorporating advanced architectural components such as SE-Res2Blocks, multi-layer feature aggregation, and channel-dependent attention mechanisms. These enhancements result in more accurate speaker embeddings, which are crucial for effective speaker verification. We explored and were not able to found any existing benchmarks for Speaker Verification in Continual learning scenario. Hence, we proposed different novel benchamrks with use of different CL methods. Experiments were performed on GPUs provided by the Cross Caps Lab, IIITD. The configuration includes Nvidia RTX 3090 having 24 GB GDDR6X memory. The GPU is operating at a frequency of 1560 MHz, which can be boosted up to 1860 MHz, memory is running at 1313 MHz (21 Gbps effective).

3.5 Results and Discussion

The ECAPA-TDNN model was tested on the VoxCeleb1, VoxCeleb1-E, VoxCeleb1-H, and VoxSRC 2019 test sets. The model demonstrated significant improvements in Equal Error Rate (EER) and minimum Detection Cost Function (MinDCF) compared to existing x-vector and ResNet-based systems. Table 3.3 shows the State of the Art Results with 0.87 EER achieved on VoxCeleb1 with 14.7 million parameters. We considered this as our comparative Baseline, however this baseline is of Traditional Methods where Dataset is stationary and the whole dataset is been passed to the model in a single pass.

The results presented represent novel approaches to speaker verification (SV) within a

Table 3.3: Baseline Results of ECAPA over VoxCeleb Dataset.

Architecture	Parameters	VoxCeleb1 EER (%)	VoxCeleb1-E EER (%)	VoxCeleb1-H EER (%)	VoxSRC19 EER (%)
E-TDNN	6.8M	1.49	1.61	2.69	1.81
E-TDNN (large)	20.4M	1.26	1.37	2.35	1.61
ResNet18	13.8M	1.47	1.60	2.88	1.97
ResNet34	23.9M	1.19	1.33	2.46	1.57
ECAPA-TDNN (C=512)	6.2M	1.01	1.24	2.32	1.32
ECAPA-TDNN (C=1024)	**14.7M**	**0.87**	**1.12**	**2.12**	**1.22**

Table 3.4: Results of Different CL strategies.

Strategy	**Data Split Type**	**EER**
Baseline	60% base + 40% in 4 stages with 10% each	52.47
Distribution based	60% high variance + 40% in 4 stages with 10% each	31.04
EWC	90% base + 10% each for each state dict of model at time t	26.84
Replay	Replay 10% of data randomly in each new split from previous split	23.39
EWC+ Replay	**Hybrid of both EWC split and Replay**	**22.46**

continual learning (CL) framework. This is the first time SV has been systematically evaluated under CL conditions, and these findings establish new baselines for the field. A comparison can be done with the Traditional method approach available in Table 3.3 with the experimental results obtained as shown in Table 3.4. Below, we explain the results and their significance:

Strategies and Results

1. **Baseline:**
   Strategy: 60% of the data is used as the base training set, followed by four stages where 10% of the data is introduced in each stage.
   EER (Equal Error Rate): **52.47**
   Explanation: This high EER indicates significant performance degradation due to catastrophic forgetting, where the model struggles to retain knowledge from earlier stages as new data is introduced.

2. **Distribution-Based Split:**
   Strategy: 60% of the data with high variance is used initially, followed by four stages with 10% data in each stage.
   EER (Equal Error Rate): **31.04**
   Explanation: Utilizing high variance data initially helps the model learn more generalizable features, leading to better performance and a lower EER compared to the baseline.

3. **Elastic Weight Consolidation (EWC):** *Strategy:* 90% of the data is used as the base set, and for each subsequent stage, the state dictionary of the model at time t is used to preserve important weights. *EER (Equal Error Rate):* **26.84**
   Explanation: EWC mitigates catastrophic forgetting by penalizing changes to parameters crucial for previously learned tasks, significantly improving performance over simpler strategies.

4. **Replay:**
   Strategy:: 10% of the data from previous stages is replayed randomly during each new split. *EER (Equal Error Rate):* **23.39**

Explanation: Replay methods help the model retain knowledge from previous stages by mixing old and new data during training, further reducing the EER.

5. **EWC + Replay (Hybrid Approach):**
   Strategy: Combines EWC's weight regularization with the Replay strategy.
   EER (Equal Error Rate): **22.46**
   Explanation: The hybrid approach leverages the strengths of both EWC and Replay, offering the best performance by ensuring important parameters are preserved while continually reinforcing old knowledge through data replay.

Significance of Results These results highlight the efficacy of various continual learning strategies in the context of speaker verification:

Novelty: This study marks the first systematic implementation of speaker verification within a continual learning framework. Each approach explored—baseline, distribution-based, EWC, Replay, and their hybrid—offers unique insights into managing catastrophic forgetting in SV systems.

Baseline Establishment: As pioneering results in this domain, these findings serve as baselines for future research. They provide a point of reference for comparing new methods and improvements in continual learning for SV.

Improved Performance: The significant reduction in EER across strategies, especially with the hybrid EWC+Replay approach, underscores the potential for continual learning techniques to enhance SV systems' robustness and accuracy over time.

By presenting these novel results, we set a foundation for ongoing advancements in speaker verification using continual learning, fostering further innovation and optimization in this critical area of research.

CHAPTER 4

Conclusion

4.1 Conclusion

The exploration of the continual learning paradigm for large-scale speaker verification systems has yielded significant insights into the potential and limitations of various strategies. Key findings and their implications are as follows:

4.1.1 Advantages:

1. **Mitigation of Catastrophic Forgetting:** Techniques like Elastic Weight Consolidation (EWC) and Replay methods effectively reduce the extent of catastrophic forgetting, enabling the model to retain previously learned information while adapting to new data.
2. **Scalability:** The proposed methods facilitate the continual integration of new speaker data without necessitating complete retraining, thus enhancing scalability.
3. **Improved Performance:** The hybrid approach combining EWC and Replay demonstrated the lowest Equal Error Rate (EER), underscoring the benefits of integrating both parameter and data-level strategies.
4. **Robustness:** Incorporating high variance data in initial training stages prepares the model to handle diverse and potentially disruptive new classes more effectively.

4.1.2 Disadvantages:

1. **Computational Complexity:** Implementing continual learning strategies, especially hybrid approaches, increases computational requirements and complexity.
2. **Memory Constraints:** Replay methods necessitate storing a subset of past data, which can pose memory constraints, particularly for very large datasets.
3. **Evaluation Challenges:** Continual learning systems require ongoing evaluation and validation to ensure that performance improvements are maintained across all learned tasks.
4. **Class Imbalance and Data Shift:** Managing class imbalance and distributional shifts remains a challenge, requiring further refinement of data splitting and handling techniques.

4.2 Summary

This thesis has presented a comprehensive study on the application of continual learning techniques to large-scale speaker verification systems. By leveraging a diverse dataset like VoxCeleb2, the research has demonstrated the feasibility and benefits of incremental learning strategies in addressing the dynamic nature of speaker verification tasks. Key contributions include:

1. **Implementation of Continual Learning Strategies:** Various strategies, including EWC, Replay, and their hybrid, were implemented and evaluated to address catastrophic forgetting and enhance model adaptability.
2. **Experimental Validation:** Extensive experiments on benchmark datasets validated the efficacy of the proposed methods, with significant improvements in EER and minimum Detection Cost Function (min DCF) observed.
3. **Development of Robust Baselines:** The research established new baselines for speaker verification under continual learning scenarios, providing a foundation for future studies.

4.3 Future Scope

The findings of this thesis open several avenues for further research and development in the field of speaker verification and continual learning:

1. **Advanced Continual Learning Techniques:** Exploring more sophisticated techniques such as meta-learning and generative replay could further enhance the adaptability and robustness of speaker verification systems.
2. **Real-Time Continual Learning:** Implementing real-time continual learning frameworks that can process streaming data without significant delays will be crucial for practical applications.
3. **Handling Imbalanced and Noisy Data:** Developing more effective methods for managing class imbalance and handling noisy data will improve system performance in real-world scenarios.
4. **Cross-Domain Adaptation:** Investigating techniques for cross-domain adaptation to enable speaker verification systems to generalize across different environments and conditions.
5. **Integration with Other Modalities:** Combining speaker verification with other biometric modalities, such as facial recognition, to enhance overall security and accuracy.

6. **Scalable Deployment:** Optimizing computational efficiency and memory usage to facilitate the scalable deployment of speaker verification systems in large-scale applications.

By addressing these areas, future research can build upon the foundations laid by this thesis to develop more robust, adaptable, and efficient speaker verification systems capable of operating in diverse and dynamic environments.

REFERENCES

1. **Andrychowicz, M.**, **M. Denil**, **S. Gomez**, **M. W. Hoffman**, **D. Pfau**, **T. Schaul**, **B. Shillingford**, and **N. De Freitas** (2016). Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, **29**.

2. **Bai, Z.** and **X.-L. Zhang** (2021). Speaker recognition based on deep learning: An overview. *Neural Networks*, **140**, 65–99. ISSN 0893-6080. URL `https://www.sciencedirect.com/science/article/pii/S0893608021000848`.

3. **Barrett, S.**, **M. E. Taylor**, and **P. Stone**, Transfer learning for reinforcement learning on a physical robot. *In Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA)*, volume 1. 2010.

4. **Bengio, Y.**, **J. Louradour**, **R. Collobert**, and **J. Weston**, Curriculum learning. *In Proceedings of the 26th annual international conference on machine learning*. 2009.

5. **Caruana, R.** (1997). Multitask learning. *Machine learning*, **28**, 41–75.

6. **Cheng, J.-M.** and **H.-C. Wang**, A method of estimating the equal error rate for automatic speaker verification. *In 2004 International Symposium on Chinese Spoken Language Processing*. IEEE, 2004.

7. **Chou, J. C.**, **C. C. Yeh**, **H. Y. Liu**, and **H. Y. Lee**, Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. *In Proceedings of the Interspeech*. 2019.

8. **Cui, Y.**, **S. Ahmad**, and **J. Hawkins** (2016). Continuous online sequence learning with an unsupervised neural network model. *Neural computation*, **28**(11), 2474–2504.

9. **Da Yu, M. Z.** (2023). Squeezing more past knowledge for online class-incremental continual learning. *IEEE/CAA Journal of Automatica Sinica*, **10**(JAS-2022-1124), 722. ISSN 2329-9266. URL `https://www.ieee-jas.net/en/article/doi/10.1109/JAS.2023.123090`.

10. **De Lange, M.**, **R. Aljundi**, **M. Masana**, **S. Parisot**, **X. Jia**, **A. Leonardis**, **G. Slabaugh**, and **T. Tuytelaars** (2022). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**(7), 3366–3385.

11. **Desplanques, B.**, **J. Thienpondt**, and **K. Demuynck**, ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. *In Proc. Interspeech 2020*. 2020. ISSN 2308-457X.

12. **Dredze, M.**, **K. Crammer**, and **F. Pereira**, Confidence-weighted linear classification. *In Proceedings of the 25th international conference on Machine learning*. 2008.

13. **Fazel, A.** and **S. Chakrabartty** (2011). An overview of statistical pattern recognition techniques for speaker verification. *IEEE Circuits and Systems Magazine*, **11**(2), 62–81.

14. **Fu, L.**, **X. Li**, **L. Zi**, **Z. Zhang**, **Y. Wu**, **X. He**, and **B. Zhou**, Incremental learning for end-to-end automatic speech recognition. *In 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021.

15. **Greenberg, C. S.**, **D. Bansé**, **G. R. Doddington**, **D. Garcia-Romero**, **J. J. Godfrey**, **T. Kinnunen**, **A. F. Martin**, **A. McCree**, **M. Przybocki**, and **D. A. Reynolds**, The nist 2014 speaker recognition i-vector machine learning challenge. *In Odyssey: The Speaker and Language Recognition Workshop*. 2014.

16. **Ha, D.**, **M. Kim**, and **C. Y. Jeong** (2023). Online continual learning in acoustic scene classification: An empirical study. *Sensors*, **23**(15). ISSN 1424-8220. URL `https://www.mdpi.com/1424-8220/23/15/6893`.

17. **He, K.**, **X. Zhang**, **S. Ren**, and **J. Sun**, Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

18. **Heo, H. S.**, **J. S. Chung**, **S. Mun**, and **A. Nagrani**, Am-mobilenet: Efficient neural networks for speaker recognition on mobile devices. *In Proceedings of the Interspeech*. 2020.

19. **Hossain, M. S.**, **P. Saha**, **T. F. Chowdhury**, **S. Rahman**, **F. Rahman**, and **N. Mohammed**, Rethinking task-incremental learning baselines. *In 2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022.

20. **Jung, J.**, **H. S. Heo**, **H. S. Shim**, and **H. J. Yu**, Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. *In Proceedings of the Interspeech*. 2019.

21. **Kaddoura, T.** and **I. Vadlamudi** (2016). Acoustic diagnosis of pulmonary hypertension: automated speech- recognition-inspired classification algorithm outperforms physicians. *Scientific Reports*, **6**.

22. **Kirkpatrick, J.**, **R. Pascanu**, **N. Rabinowitz**, **J. Veness**, **G. Desjardins**, **A. A. Rusu**, **K. Milan**, **J. Quan**, **T. Ramalho**, **A. Grabska-Barwinska**, **D. Hassabis**, **C. Clopath**, **D. Kumaran**, and **R. Hadsell** (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, **114**(13), 3521–3526. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1611835114`.

23. **Kivinen, J.**, **A. J. Smola**, and **R. C. Williamson** (2004). Online learning with kernels. *IEEE transactions on signal processing*, **52**(8), 2165–2176.

24. **Kreuk, F.**, **S. Chazan**, **Y. Adi**, and **J. Keshet**, Fooling end-to-end speaker verification with adversarial examples. *In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018.

25. **Li, Y.**, **W. Zhang**, **X. Xu**, **Y. He**, **D. Dong**, **N. Jiang**, **F. Wang**, **Z. Guo**, **S. Wang**, **C. Dou**, **Y. Liu**, **Z. Wang**, and **D. S. Shang** (2022). Mixed-precision continual learning based on computational resistance random access memory. *Advanced Intelligent Systems*, **4**.

26. **Liu, B.**, Learning on the job: Online lifelong and continual learning. *In Proceedings of the AAAI conference on artificial intelligence*, volume 34. 2020.

27. **Liu, H.**, **Y. Zhou**, **B. Liu**, **J. Zhao**, **R. Yao**, and **Z. Shao** (2023). Incremental learning with neural networks for computer vision: a survey. *Artificial Intelligence Review*, **56**(5), 4557–4589.

28. **Lomonaco, V.** (2019). Continual learning with deep architectures.

29. **Long, M.**, **H. Zhu**, **J. Wang**, and **M. I. Jordan**, Deep transfer learning with joint adaptation networks. *In* **D. Precup** and **Y. W. Teh** (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017. URL `https://proceedings.mlr.press/v70/long17a.html`.

30. **Mairal, J.**, **F. Bach**, **J. Ponce**, and **G. Sapiro** (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, **11**(1).

31. **Mittal, A.** and **M. Dua** (2022). Automatic speaker verification systems and spoof detection techniques: review and analysis. *International Journal of Speech Technology*, **25**.

32. **Nagrani, A.**, **J. S. Chung**, and **A. Zisserman**, VoxCeleb: A Large-Scale Speaker Identification Dataset. *In Proc. Interspeech 2017*. 2017. ISSN 2308-457X.

33. **Ohi, A. Q.**, **M. F. Mridha**, **M. A. Hamid**, and **M. M. Monowar** (2021). Deep speaker recognition: Process, progress, and challenges. *IEEE Access*, **9**, 89619–89643.

34. **Pan, S. J.** (2020). Transfer learning. *Learning*, **21**, 1–2.

35. **Parisi, G. I.** and **V. Lomonaco**, Online continual learning on sequences. *In Recent Trends in Learning From Data: Tutorials from the INNS Big Data and Deep Learning Conference (INNSBDDL2019)*. Springer, 2020.

36. **Pellegrini, L.**, **G. Graffieti**, **V. Lomonaco**, and **D. Maltoni** (2019). Latent replay for real-time continual learning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10203–10209. URL `https://api.semanticscholar.org/CorpusID:208547551`.

37. **Pelosin, F.** (2023). Dissecting continual learning a structural and data analysis.

38. **Rabiner, L.** and **B. H. Juang**, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

39. **Ravanelli, M.** and **Y. Bengio**, Speaker recognition from raw waveform with sincnet. *In 2018 IEEE Spoken Language Technology Workshop (SLT)*. 2018.

40. **Reynolds, D. A.** and **R. C. Rose** (1995). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, **3**(1), 72–83.

41. **Snell, J.**, **K. Swersky**, and **R. Zemel**, Prototypical networks for few-shot learning. *In Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017.

42. **Snyder, D.**, **D. Garcia-Romero**, **G. Sell**, **D. Povey**, and **S. Khudanpur**, X-vectors: Robust dnn embeddings for speaker recognition. *In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018.

43. **Sun, S.**, **Y. Qian**, and **H. Liu**, A hybrid domain adaptation approach for robust speaker verification. *In Proceedings of the Interspeech*. 2017.

44. **Sutskever, I.**, **O. Vinyals**, and **Q. V. Le** (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, **27**.

45. **Sutton, R. S.**, Open theoretical questions in reinforcement learning. *In European Conference on Computational Learning Theory*. Springer, 1999.

46. **Taylor, M. E.** and **P. Stone** (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, **10**(7).

47. **Van de Ven, G. M.**, **T. Tuytelaars**, and **A. S. Tolias** (2022). Three types of incremental learning. *Nature Machine Intelligence*, **4**(12), 1185–1197.

48. **Van Otterlo, M.** and **M. Wiering**, Reinforcement learning and markov decision processes. *In Reinforcement learning: State-of-the-art*. Springer, 2012, 3–42.

49. **Variani, E.**, **X. Lei**, **E. McDermott**, **I. L. Moreno**, and **J. Gonzalez-Dominguez**, Deep neural networks for small footprint text-dependent speaker verification. *In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014.

50. **Yang, M.** (2023). *Continual Learning in Speech and Audio Applications*. Ph.D. thesis, CARNEGIE MELLON UNIVERSITY.

51. **Yang, M.**, **I. Lane**, and **S. Watanabe** (2022). Online continual learning of end-to-end speech recognition models. *arXiv preprint arXiv:2207.05071*.

52. **Zhu, Y.** (2022). *Deep speaker representation learning in speaker verification*. Ph.D. thesis, Hong Kong University of Science and Technology.